|  |  |
| --- | --- |
| **Desarrollo de Aplicaciones en Entorno Cliente**  **2º Ciclo Superior de Desarrollo de Aplicaciones Web**  **2023-2024** |  |
| **React** |  |

**Tareas**

1. Crea un componente React que muestre una página con tres componentes funcionales: **Encabezado**, **Contenido** y **PieDePagina**. El componente **Encabezado** debe mostrar un título, **Contenido** debe mostrar un párrafo de texto y **PieDePagina** debe mostrar el año actual. Combina estos componentes en un componente principal llamado **Pagina** y muestra todos los componentes en la página. A continuación se muestra el archivo *Pagina.jsx* que tienes que completar:

// Pagina.jsx

import React from 'react';

import Encabezado from './Encabezado';

import Contenido from './Contenido';

import PieDePagina from './PieDePagina';

function Pagina() {

return (

<div>

<Encabezado />

<Contenido />

<PieDePagina />

</div>

);

}

export default Pagina;

1. Crea un componente React que muestre una lista de nombres de usuarios. Para ello, realiza los siguientes componentes (han de ser reutilizados uno dentro de otro):
   1. Crea un componente **Usuario** que reciba un nombre de usuario y genere una etiqueta <li> con dicho nombre de usuario.
   2. Crea un componente **ListaUsuarios** que tome una matriz de nombres como propiedad y muestre cada nombre en un componente **Usuario**. Asegúrate de que los nombres se muestren en una lista ordenada (<ol>).
   3. Crea por último, el componente principal llamado **MuestraListaUsuarios** que declara el array con los nombres de usuarios y utiliza el componente **ListaUsuarios** para mostrar la lista de ese array creado.
2. Crea una aplicación que sirva para visualizar la solución de los ejercicios anteriores y de los que posteriores que hagamos en esta Unidad Didáctica.  
   Para ello, tienes que colocar en la cabecera varios botones (uno por cada solución) que hagan visible o invisible a los componentes que vayamos creando en cada ejercicio (ya los puedes hacer para los dos anteriores). Por defecto, todos los componentes con las soluciones de cada ejecución, aparecerán no visibles.  
   (Pistas: puedes utilizar el atributo hidden con un estado del componente principal de la aplicación para controlar la visibilidad y los botones pueden llamar a eventos que cambien el estado del componente principal:  
   const [contenedor1, setContenedor1] = useState(false);  
   setContenedor1(!contenedor1);  
   <div id="id1" hidden={!contenedor1}><Componente1 /></div>
3. Crea un componente React que conste de un botón y un cuadro de texto.  
   Cuando el usuario haga clic en el botón, el cuadro de texto debe mostrar el mensaje "¡Hiciste clic en el botón!" y cambiar su color de fondo a verde.  
   Cuando el usuario pase el cursor sobre el cuadro de texto, el color de fondo debe cambiar a amarillo.  
   Cuando el usuario retire el cursor del cuadro de texto, el color de fondo debe volver a ser blanco.
4. Crea una componente React que conste de dos componentes: ContadorA y ContadorB:
   1. Cada componente debe incluir un botón y un contador.
   2. Cuando el usuario haga clic en el botón del ContadorA, su contador debe aumentar en 1.
   3. Cuando el usuario haga clic en el botón del ContadorB, su contador debe aumentar en 2.
   4. Cada componente debe llevar su propio estado independiente.
5. Modifica el componente para que el estado de los contadores (contadorA y contadorB) debe ser compartido entre los dos componentes. De esta manera, cuando se haga clic en los botones de cualquiera de los dos contadores, ambos contadores deben actualizarse según la lógica original.